CLAIMS

What is claimed is:

1. An apparatus comprising:

a processor core including data paths;

a plurality of peripheral devices having associated parallel scan registers coupled to the processor core; and

operating logic for generating pseudo-random test patterns for the peripheral devices, employing a mixed congruential generation scheme, and using the data paths of the processor core.

2. The apparatus as set forth in claim 1, wherein the operating logic includes operating logic for multiplying n least significant bits of a 2n-bit pseudo-random number generated in an immediately preceding iteration and stored in a first register, with an n-bit multiplier constant stored in a second register to produce a 2n-bit product, adding the 2n-bit product to n most significant bits of the 2n-bit pseudo-random number stored in n least significant locations of an accumulator with 2n locations to produce a new 2n-bit pseudo-random number for a current iteration, and outputting n least significant bits of the new 2n-bit pseudo-random number as an n-bit pseudo-random test vector for the peripheral devices.

3. The apparatus as set forth in claim 2, wherein the operating logic further includes operating logic for copy the n least significant bits of the new 2n-bit pseudo-random number into the first register, and right shifting the 2n-bit new pseudo-random number in the accumulator by n-bits.

4. The apparatus as set forth in claim 3, wherein the operating logic further includes operating logic for repeating said multiplying, adding, outputting, copying and right shifting sufficient number of times to generate a sufficiently large pseudo-random test pattern to fill the parallel scan registers.

5. The apparatus as set forth in claim 2, wherein the operating logic further includes operating logic for storing a 2n-bit initial value into the accumulator, and storing the n-bit multiplier constant in the second register.

6. The apparatus as set forth in claim 5, wherein the variable n, the 2n-bit initial value and the n-bit multiplier constant are equal to (3, 6, 1), (3, 6, 5), (4, 15, 1), (4, 15, 7), (5, 27, 1), (5, 27, 5), (6, 45, 1), (6, 45, 7), (7, 126, 1), (7, 126, 5), (8, 249, 1), (8, 249, 5), (9, 507, 1) (9, 507, 5), (10, 1020, 1), (10, 1020, 5), (11, 2016, 1), (11, 2016, 5), (12, 4077, 1), (12, 4077, 7), (13, 8175, 1), (13, 8175, 7), (14, 16371, 1), (14, 16371, 5), (15, 32766, 1), (15, 32766, 5), (16, 65514, 1), or (16, 65514, 5) respectively.

7. The apparatus as set forth in claim 2, wherein the apparatus further comprises a test port register through which the parallel scan registers are coupled to the processor core, and the operating logic further includes operating logic for moving a pseudo-random test vector from the processor core to the test port register, and then moving the pseudo-random test vector in parallel to the parallel scan registers.

8. An apparatus comprising:

a processor core including data paths;

a plurality of peripheral devices having associated parallel scan registers coupled to the processor core; and operating logic for generating deterministic test patterns for the peripheral devices, emulating a plurality of linear feedback shift register (LFSR) - based segments interconnected by a network of linear functions, and using the data paths of the processor core.

9. The apparatus as set forth in claim 8, wherein the operating logic includes operating logic for modifying n x L storage locations of a circular buffer formed with L registers with n storage locations each of processor core, emulating shifting of the content of the LFSR-based segments, and outputting the content of a head one of the L registers as a portion of a deterministic test pattern being generated for the peripheral devices.

10. The apparatus as set forth in claim 9, wherein the operating logic further includes operating logic for repeating said modification and outputting sufficient number of times to generate a sufficiently large deterministic test pattern to fill the parallel scan registers.

11. The apparatus as set forth in claim 9, wherein the operating logic further includes operating logic for advancing a head pointer for identifying the head one of the L registers.

12. The apparatus as set forth in claim 8, wherein the operating logic generates the deterministic test patterns emulating LFSR-based segments interconnected with a network of linear functions that implements an identical feedback polynomial for each of the LFSR-based segments.

13. The apparatus as set forth in claim 8, wherein the operating logic generates the deterministic test patterns emulating LFSR-based segments interconnected with a

network of linear functions that provides inter-segment feedback at identical tap positions of the LFSR-based segments.

14. The apparatus as set forth in claim 8, wherein the apparatus further comprises a test port register through which the parallel scan registers are coupled to the processor core, and the operating logic further includes operating logic for moving a deterministic test vector from the processor core to the test port register, and then moving the pseudo-random test vector in parallel to the parallel scan registers.

15. An apparatus comprising:

a processor core including data paths;

a plurality of peripheral devices having associated parallel scan registers coupled to the processor core; and

operating logic for compacting test responses the peripheral devices, employing a cascaded approach, and using the data paths of the processor core.

16. The apparatus as set forth in claim 15, wherein the operating logic includes operating logic for moving a first test response into a first accumulator, adding a first signature to the first accumulator, moving a second test response into a second accumulator, outputting the content of the first accumulator and adding the output of the first accumulator into the second accumulator.

17. The apparatus as set forth in claim 16, wherein the operating logic further includes operating logic for moving the second test response into the first accumulator, adding a second signature that includes the first test response into the first accumulator, moving a third test response into the second accumulator, outputting the content of the first accumulator, and adding the output of the first accumulator to the second accumulator.

18. The apparatus as set forth in claim 17, wherein the operating logic adds the output of the first accumulator into the second accumulator in accordance with a 1's complement convention.

19. The apparatus as set forth in claim 17, wherein the operating logic adds the output of the first accumulator into the second accumulator using a rotate carry scheme.

20. The apparatus as set forth in claim 15, wherein the apparatus further comprises a test port register through which the parallel scan registers are coupled to the processor core, and the operating logic further includes operating logic for moving a test response of the peripheral devices from the test port register to the processor core.

21. The apparatus as set forth in claim 20, wherein the test response of the peripheral devices is recovered in the test port register as an integral part of storing a test vector in parallel into the parallel scan registers.

22. An apparatus comprising:

a processor core including data paths;

a plurality of peripheral devices having associated parallel scan registers; and

a test port register coupled to the processor core and the parallel scan registers for loading the parallel scan registers with test patterns generated by the data paths of the processor core for testing the peripheral devices, and for recovering test responses of the peripheral devices to application of the test patterns from the parallel scan registers for the processor core.

23. The apparatus as set forth in claim 22, wherein the test port register comprises a plurality of output ports correspondingly coupled to first cells of the parallel scan registers, and a plurality of input ports correspondingly coupled to last cells of the parallel scan registers.

24. An apparatus comprising:

a processor core including data paths;

a plurality of peripheral devices having associated parallel scan registers;

a test port register coupled to the processor core and the parallel scan registers; and

operating logic for moving a test pattern generated by the data paths of the processor core to the test port register, and then to the parallel scan registers.

25. The apparatus as set forth in claim 24, wherein the operating logic further includes operating logic for applying one or more clock cycles to the peripheral devices to apply the test pattern in the parallel scan registers to the peripheral devices.

26. The apparatus as set forth in claim 24, wherein the operating logic further includes operating logic for moving test responses of the peripheral devices from the test port register to the processor core.

26. An apparatus comprising:

a processor core including data paths;

a plurality of peripheral devices having associated parallel scan registers coupled to the processor core; and

operating logic for placing the processor core in a test mode, saving a pre-test state of the parallel scan registers, applying a plurality of test patterns to the peripheral devices using the data paths of the processor core, and restoring the pre-test state of the parallel scan registers.

27. The apparatus as set forth in claim 26, wherein the operating logic for applying a plurality of test patterns to the peripheral devices includes operating logic for generating one or more test vectors for the peripheral devices using the data paths of the processor core, and compacting test responses of the peripheral devices using the data paths of the processor core.

28. An apparatus comprising:

a processor core including data paths;

a plurality of peripheral devices coupled to the processor core; and

operating logic for generating one or more test vectors for the peripheral devices using the data paths of the processor core, and compacting test responses of the peripheral devices using the data paths of the processor core.

29. A method for generating pseudo-random test patterns comprising the steps of:

multiplying n least significant bits of a 2n-bit pseudo-random number generated in an immediately preceding iteration and stored in a first register, with an n-bit multiplier constant stored in a second register to produce a 2n-bit product,

adding the 2n-bit product to n most significant bits of the 2n-bit pseudo-random number stored in n least significant locations of an accumulator with 2n locations to produce a new 2n-bit pseudo-random number for a current iteration, and

outputting n least significant bits of the new 2n-bit pseudo-random number as an n-bit pseudo-random test vector for the peripheral devices.

30. The method as set forth in claim 29, wherein the method further includes the step of copying the n least significant bits of the new 2n-bit pseudo-random number into the first register, and right shifting the 2n-bit new pseudo-random number in the accumulator by n-bits.

31. The method as set forth in claim 30, wherein the method further includes the step of repeating said multiplying, adding, outputting, copying and right shifting steps sufficient number of times to generate a sufficiently large pseudo-random test pattern to fill a parallel scan registers.

32. The method as set forth in claim 29, wherein the method further includes the step of storing a 2n-bit initial value into the accumulator, and storing the n-bit multiplier constant in the second register.

33. The method as set forth in claim 32, wherein the variable n, the 2n-bit initial value and the n-bit multiplier constant are equal to (3, 6, 1), (3, 6, 5), (4, 15, 1), (4, 15, 7), (5,

27, 1), (5, 27, 5), (6, 45, 1), (6, 45, 7), (7, 126, 1), (7, 126, 5), (8, 249, 1), (8, 249, 5), (9, 507, 1) (9, 507, 5), (10, 1020, 1), (10, 1020, 5), (11, 2016, 1), (11, 2016, 5), (12, 4077, 1), (12, 4077, 7), (13, 8175, 1), (13, 8175, 7), (14, 16371, 1), (14, 16371, 5), (15, 32766, 1), (15, 32766, 5), (16, 65514, 1), or (16, 65514, 5) respectively.

34. The method as set forth in claim 29, wherein the method further comprises the step moving a pseudo-random test vector to a test port register, and then moving the pseudo-random test vector in parallel to a plurality of parallel scan registers.

35. A method for generating deterministic test patterns comprising the steps of:

modifying n x L storage locations of a circular buffer formed with L registers, each having n storage locations,

emulating shifting of the content of a plurality of LFSR-based segments, and

outputting the content of a head one of the L registers as a portion of a deterministic test pattern being generated.

36. The method as set forth in claim 35, wherein the method further includes the step of repeating said modification and outputting sufficient number of times to generate a sufficiently large deterministic test pattern to fill a plurality of parallel scan registers.

37. The method as set forth in claim 35, wherein the method further includes the step of advancing a head pointer for identifying the head one of the L registers.

38. The method as set forth in claim 35, wherein the emulation step emulates shifting of LFSR-based segments interconnected with a network of linear functions that implements an identical feedback polynomial for each of the LFSR-based segments.

39. The method as set forth in claim 35, wherein the emulation step emulates shifting of LFSR-based segments interconnected with a network of linear functions that provides inter-segment feedback at identical tap positions of the LFSR-based segments.

40. The method as set forth in claim 35, wherein the method further comprises the steps of moving a deterministic test vector to a test port register, and then moving the pseudo-random test vector in parallel to a plurality of parallel scan registers.

41. A method comprising the steps of:

moving a first test response into a first accumulator,
adding a first signature to the first accumulator,
moving a second test response into a second accumulator,
outputting the content of the first accumulator and
adding the output of the first accumulator into the second accumulator.

42. The method as set forth in claim 41, wherein the method further includes the step of:

moving the second test response into the first accumulator,
adding a second signature that includes the first test response into the first accumulator,
moving a third test response into the second accumulator,
outputting the content of the first accumulator, and
adding the output of the first accumulator to the second accumulator.

43. The method as set forth in claim 42, wherein the step of adding of the output of the first accumulator into the second accumulator is performed in accordance with a 1's complement convention.

44. The method as set forth in claim 42, wherein the step of adding the output of the first accumulator into the second accumulator is performed using a rotate carry scheme.

45. The method as set forth in claim 41, wherein the method further comprises the step of moving a partial test response from a test port register.

46. The method as set forth in claim 45, wherein the step of moving the partial test response is performed as an integral part of a step of providing a test vector from the test port register.

47. A method for testing peripheral devices of an integrated circuit, the method comprising the steps of:

generating first test patterns using mission datapaths of a processor core;

loading a plurality of parallel scan registers with the first test patterns from a test port register, one bit slice for each scan register at a time, and simultaneously recovering test responses for previously generated second test patterns from the parallel scan registers into the test port register, one bit slice per scan register a time;

moving the test responses into the processor core, and compacting the test responses into a signature.

48. The method as set forth in claim 47, wherein the method further includes the step of applying one or more clock cycles to the peripheral devices to apply the first test patterns in the parallel scan registers to the peripheral devices.

49. A method for testing an integrated circuit, comprising the steps of:

placing a processor core in a test mode;

saving a pre-test state of a plurality of parallel scan registers,

generating and applying a plurality of test patterns to a plurality of peripheral devices using the data paths of the processor core and the parallel scan registers, and

restoring the pre-test state of the parallel scan registers.

50. The method as set forth in claim 49, wherein the generating and applying step includes generating one or more test vectors for the peripheral devices using the data paths of the processor core, and compacting test responses of the peripheral devices using the data paths of the processor core.